Inteligencia Artificial Avanzada:

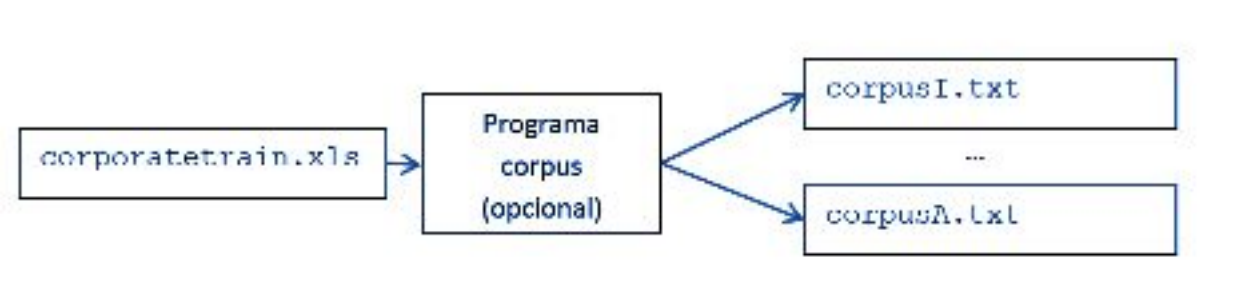
Práctica: Clasificación de Textos en Lenguaje Natural

Angel Luis Igareta Herráiz

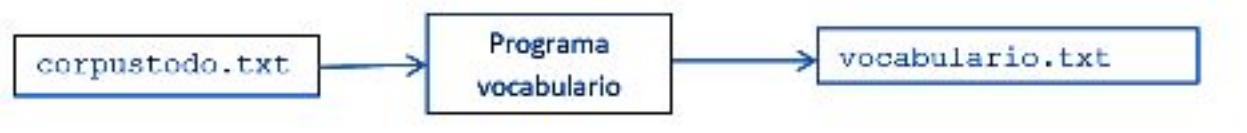
Miguel Jiménez Gomis

PARTE 1

Los primeros pasos realizados para la realización de este trabajo fue la generación de los diferentes corpus. Este proceso consistió en separar las diferentes categorías que se encuentran en el corpus general (Acción, Diálogo e Información) en un corpus por cada categoría.



El siguiente paso consiste en generar el vocabulario. Este se trata de un fichero que contiene todas las palabras que nos podemos encontrar en los diferentes corpus pero sin duplicados. Además, el vocabulario está ordenado alfabéticamente.



Cabe añadir que tuvimos que realizar una tarea extra, la de **limpieza de datos,** ésto se debe a que, en el corpus que se nos daba, había caracteres extraños y algunos tokens que vimos que para clasificación no nos serían de utilidad. Es por ello que hicimos varias sustituciones y eliminaciones de caracteres extraños hasta que vimos que el resultado era el correcto.

Además, inspirándose de MonkeyLearn, añadimos dos tokens: **_number_** y **_url_**, y los añadimos cada vez que veíamos un número o una dirección http, respectivamente. Esto nos ayudará en procesos posteriores a clasificar de una mejor manera el corpus que se introduzca y a no sobre ajustar tanto el vocabulario obtenido.

Inicialmente realizamos este proceso en dos programas separados, en java la concatenación y limpieza de los corpus y en python la ordenación y eliminación de palabras repetidas para la generación del vocabulario (Se hicieron en lenguajes diferentes debido a las facilidades que tienen dichos lenguajes para realizar esas tareas).

Más adelante para una mejor integración en el código se desarrolló la clase Parser en Java y se prescindió del programa de python. En esta clase se convierte un documento en tokens, utilizando la sustitución y limpieza anteriormente explicada.

PARTE 2

La siguiente tarea que realizamos fue la estimación de probabilidades para generar los ficheros de aprendizaje. Básicamente el esquema de el programa a realizar es el siguiente, dónde como entrada tenemos los corpus y el vocabulario generados en la parte 1:

Cabe destacar que los ficheros de aprendizaje constan de un formato en especial. Tienen dos cabeceras en las que se indica el número de palabras del corpus del que estamos haciendo el fichero de aprendizaje y el número de palabras del vocabulario.

A continuación le sigue una lista en la que, para cada palabra del vocabulario se representa la palabra, su frecuencia en el corpus del que se está "aprendiendo", y el logaritmo de su probabilidad. Se hace ésto frente sólo a calcular la probabilidad nada más porque, al hacer el logaritmo, aporta un rango de variación más amplio.

A su vez, a la probabilidad se le hace un suavizado laplaciano para poder tratar con palabras de frecuencia 0, pues puede darse el caso de haber palabras en el vocabulario que no se encuentren en el corpus. La probabilidad se calcula con la fórmula:

$$\textbf{P (palabra) =} \frac{frecuencia\ de\ la\ palabra + 1}{número\ de\ palabras\ del\ corpus + número\ de\ palabras\ del\ vocabulario}$$

Para realizar ésta parte del programa se añadieron dos nuevas clases:

- **Token**: Representa una palabra o token del corpus. Esta tiene un identificador y una frecuencia, que variará dependiendo del corpus en el que esté.

- **Corpus**: Contiene un array ordenado de tokens y su frecuencia. En el constructor se le pasa el vocabulario y por cada una de las palabras que contiene se le asigna las probabilidades de la forma que hemos expuesto. A su vez contiene el número de palabras del vocabulario y del corpus, así como la probabilidad del corpus, para no hacer cálculos redundantes.

PARTE 3

La última tarea para completar la práctica fue desarrollar un programa que, dado una serie de documentos (en nuestro caso tweets) los clasifique en el corpus al que pertenecen. Es decir, si nos dan un tweet decir si es de la categoría Acción, Información o Diálogo.

En el siguiente esquema podemos ver que el programa a desarrollar debe recibir como entrada los ficheros de aprendizaje y el corpus a clasificar. Además, como salida, debe mostrar cada documento del corpus con la categoría a la que pertenece.

Para realizar la clasificación de un documento, debemos calcular la probabilidad de que aparezca ese documento en cada uno de los corpus y quedarnos con la que sea mayor. Para calcular la probabilidad de un documento, deberemos calcular la probabilidad de cada una de las palabras y sumarlas entre sí, pues estamos trabajando con logaritmos. Se hará de la siguiente manera:

P (documento) = $\log P(word1|corpus) + \log P(word2|corpus) + \ldots + \log P(word_n|corpus) + \log P(corpus)$

Con el fin de realizar la clasificación hemos hecho una clase llamada Classifier, que recibe un documento y, utilizando los corpus que hemos generado anteriormente, calcula la probabilidad

de la manera que se acaba de exponer. Finalmente devuelve una cadena con el nombre de la categoría a la que pertenece, Acción, Diálogo e Información.

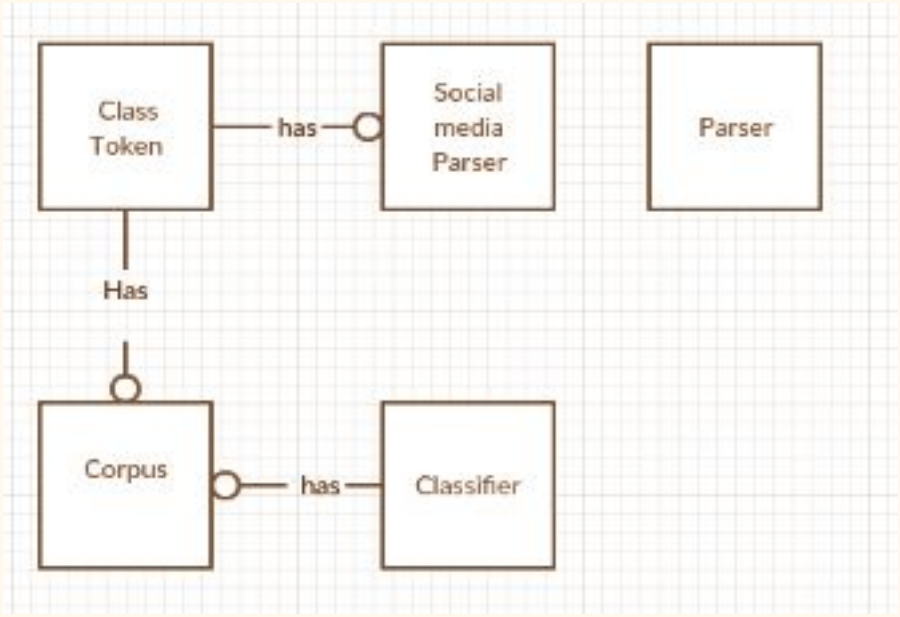
A medida que vamos clasificando, vamos contando el número de aciertos y escribiendo en el fichero “clasificación.txt” la categoría y el documento, en formato CSV.

Probándolo sobre el corpus general de la primera práctica, nos da un **porcentaje de 84.56%** documentos bien clasificado. Hemos conseguido llegar al mismo al utilizar tokens y una limpieza inteligente.

DETALLES DEL CÓDIGO

Cabe añadir que todo el código que se ha usado en la práctica está disponible en: https://github.com/alu0100970876/socialmediadata

Finalmente en esta imagen se puede ver un diagrama de las clases que se han implementado para la realización de este programa y que se han explicado anteriormente, así cómo están relacionadas entre ellas.



PARTICIPACIÓN

Miguel 40%, Angel 60%